



## Laboratório 5 - Tabelas de Dispersão

Semana de 21 a 25 de Novembro de 2017      Duração: 2 horas

O objectivo deste trabalho é o estudo de uma implementação de uma tabela de dispersão (hash-table), usando o método de separação em listas. A tabela será usada para armazenar palavras.

O slides sobre Tabelas de Dispersão que acompanham este laboratório explicam o funcionamento genérico desta estrutura de dados (slides 1 a 11) e a forma como é implementada com resolução de colisões por separação em listas (slides 20 a 24). No início da laboratório o Docente fará uma breve exposição sobre a representação de dados em Tabelas de Dispersão e as operações envolvidas.

O programa, a completar na aula, lê as palavras contidas num ficheiro e coloca-as na tabela de dispersão juntamente com o seu número de ocorrências. Após a construção da tabela, o seu conteúdo (palavras distintas e respectivo número de ocorrências) é escrito num ficheiro `<nome_ficheiro_entrada>.palavras`.

Os ficheiros `hash.c` e `pal.c` contêm código, designadamente a definição das estruturas e funções indicadas abaixo. Análise o código que lhe é fornecido. O programa recebe 2 argumentos na linha de comando: o nome do ficheiro de entrada e o valor do parâmetro  $m$ .

- O código implementa uma estrutura de dados em lista, onde cada elemento contém uma referência para uma cadeia de caracteres e um inteiro para guardar o número de ocorrências.
- A função `AlocaNo(...)` recebe uma cadeia de caracteres, aloca e inicializa o nó.
- A função `AcréscentaNo(no1, no2)` tem argumentos do tipo apontador para No. A função acrescenta o nó apontado por `no1` à lista, colocando-o a seguir ao nó apontado por `no2`.
- A função `AlocaTabela(...)` tem um argumento do tipo inteiro, que corresponde ao número de elementos ( $m$ ) da tabela. Os elementos da tabela são de tipo apontador para No.
- A função de dispersão `Disperse(...)`, aceita um argumento de tipo cadeia de caracteres (palavra) e um argumento do tipo inteiro ( $m$ ). A função devolve um inteiro entre 0 e  $m - 1$  que será usado com índice na tabela de dispersão. O valor é calculado através do seguinte pseudocódigo:

```
Disperse(palavra, m)
    h = 0
    PESO = 117
    n = comprimento(palavra)
    para i entre 0 e n-1
        executar h = mod(palavra[i] * PESO + h, m)
    devolver h
```

- A função `InsererNaTabela(...)` insere uma cadeia de caracteres (passada em argumento) na tabela, de acordo com o índice produzido pela função de dispersão. Se a palavra já se encontrar na tabela, não é novamente inserida, mas o número de ocorrências é incrementado. Caso a palavra não exista na tabela, mas a função de dispersão forneça um índice já ocupado na tabela (ocorrência de colisão de indexação), insere o novo nó (com a nova palavra) no fim da lista.

1. As funções de leitura e escrita dos ficheiros são fornecidas no ficheiro `pal.c`. Modifique e acrescente o código necessário na função `MostraDistribuicao(...)` para actualizar os dados da estrutura `Info` com os valores que indicam o tamanho da maior lista e o número de listas vazias da tabela de dispersão.

## 2. Resolver e folha para entregar no fim do laboratório

Experimente o seu programa com os ficheiros de texto que se encontram na sua área. Verifique o comprimento das listas criadas e o número de comparações de cadeias de caracteres efectuadas durante a inserção das palavras na tabela, para vários valores do parâmetro  $m$ :

- $m = 1$
- $m = 11$
- $m = 13$
- $m = 120$
- $m = 117$
- $m = 121$
- $m = 128$

Avalie e justifique os resultados.

3. Usando o `doc2.txt` para os seguintes pares de  $PESO$  e  $m$ , determine e registe o tamanho da lista maior e o número de listas vazias. Se a função de dispersão fosse ideal quais deveriam ser esses valores? Justifique brevemente as diferenças encontradas.

- $PESO = 153, \quad m = 255$
- $PESO = 153, \quad m = 257$
- $PESO = 765, \quad m = 12,$

4. Para os dois primeiros pares de  $m$  e  $PESO$  quantas comparações em média seriam necessárias para procurar uma palavra que exista e outra que não exista na tabela de dispersão?